

NE-1050

- 1 -

1 TITLE OF THE INVENTION

2 **Method and System for Client-Server Communication**

3 BACKGROUND OF THE INVENTION

4 Field of the Invention

5 The present invention relates to a client-server system which allows
6 client's processors to identify a server on a real-time basis even when data file
7 is moved from one server to another.

8 Description of the Related Art

9 In the usual plug-in environment in which an external device such as
10 an HTML (hypertext markup language) browser of a client terminal fetches a
11 file from a Web server and uses it to invoke the processor of the client
12 terminal, the file acquisition process is fully entrusted on the browser. As a
13 result, when accessing the server, the invoked client's processor must use the
14 same URL (uniform resource locator) as one that was used by the browser to
15 access the server. However, the ability to ascertain whether the URL
16 specified by the browser is actually available depends on the functions of the
17 browser. As a result, when the client terminal were to sequentially access
18 different servers, the technique for identifying the servers would become too
19 complex to be implemented.

20 Another method involves writing an URL in a file to be downloaded
21 and using the URL to acquire a different file. However, browser's cache
22 control cannot completely be performed by a processor of the type which is
23 invoked after a file is downloaded. Further, information sharing on a real-
24 time basis cannot consistently be provided. Therefore, it is difficult to
25 guarantee that files which a browser downloads are of most recent origin. In

09870809-053101

NE-1050

- 2 -

1 addition, when the client terminal changes servers, it needs to alter the URI.
2 of the server information in the file, which represents a time-consuming job.
3 This is a critical problem in a private server environment in which the server
4 is frequently moved from one location to another. Thus, when a server is
5 altered, data cannot simply be transferred.

6 The approach that is most extensively used as an expedient method is
7 one in which server information is saved in the client terminal. When the
8 client terminal accesses a server, it uses the stored information. However, if
9 the client terminal were to sequentially access different servers, a complex
10 technique would be required to dynamically change servers.

11 SUMMARY OF THE INVENTION

12 It is therefore an object of the present invention to provide a method
13 and a system for allowing client's processors to identify a server on a real-
14 time basis even when data file is moved from one server to another.

15 According to a first aspect of the present invention, there is provided a
16 method of identifying a server from a client terminal having a browser and a
17 processor, the server and the client terminal being connectable with each
18 other via a communications network, comprising the steps of (a) transmitting
19 a first request packet from the browser to the server, (b) receiving the first
20 request packet at the server and transmitting therefrom server specific
21 information to the browser, indicating a server in which shared data file is
22 maintained, (c) receiving the server specific information at the browser and
23 invoking the processor to hand over the received information to the
24 processor, (d) transmitting a second request packet from the processor to a
25 server specified by the received server specific information, and (e) receiving

TOE50-50802860

NE-1050

- 3 -

1 the second request packet at the specified server and transmitting therefrom
2 the shared data file to the processor.

3 According to a second aspect, the present invention provides a method
4 of identifying a server from a client terminal having a browser and a
5 processor, the server and the client terminal being connectable with each
6 other via a communications network. The method comprises the steps of (a)
7 transmitting a first request packet from the browser to the server, (b)
8 receiving the first request packet at the server and transmitting therefrom
9 server specific information to the browser, indicating a server in which
10 shared data file is maintained, (c) receiving the server specific information at
11 the browser and storing the received information, (d) invoking the processor
12 and transmitting a second request packet therefrom to a server specified by
13 the stored server specific information, and (e) receiving the second request
14 packet at the specified server and transmitting therefrom the shared data file
15 to the processor.

16 According to a third aspect, the present invention provides a client-
17 server system comprising a communications network, a server connected to
18 the network, and a client terminal connected to the network, the client
19 terminal having a processor and a browser, the browser transmitting a first
20 request packet to the server. The server responds to the first request packet
21 by transmitting server specific information to the browser for indicating a
22 server in which shared data file is maintained. The browser is responsive to
23 the server specific information for invoking the processor to hand over the
24 received information. The processor is responsive to the received
25 information for transmitting a second request packet to a server specified by
26 the received information. The specified server is responsive to the second

TOPEO 60802860

NE-1050

- 4 -

1 request packet for transmitting the shared data file to the processor.
2 According to a fourth aspect, the present invention provides a client-
3 server system comprising a communications network, a server connected to
4 the network, and a client terminal connected to the network, the client
5 terminal having a processor and a browser, the browser transmitting a first
6 request packet to the server. The server is responsive to the first request
7 packet for transmitting a server specific information to the browser for
8 indicating a server in which shared data file is maintained. The browser
9 receives the server specific information and stores it and invokes the
10 processor. The processor reads the stored information and transmits a
11 second request packet to a server specified by the stored information. The
12 specified server is responsive to the second request packet for transmitting
13 the shared data file to the processor.

14 If the shared data file were transferred from the first server to a second
15 server, the server specific information would be updated by the first server so
16 that the client terminal can access the new server to obtain the data file.

17 BRIEF DESCRIPTION OF THE DRAWINGS

18 The present invention will be described in detail further with reference
19 to the following drawings, in which:

20 Fig. 1 is a block diagram of a client-server system of the present
21 invention; and

22 Fig. 2 is a flowchart of the operation of the system.

23 DETAILED DESCRIPTION

24 A client-server system of the present invention shown in Fig. 1
25 comprises a plurality of client terminals 100 connected to the Internet
26 network 110 which includes proxy servers 111. Through the network, each

TO: F50, 6080/860

NE-1050

- 5 -

1 client terminal can access one of a plurality of Web servers 120-1 and 120-2.
2 Each client terminal includes a processor 101, a browser program 102, and a
3 memory device 103 for storing server specific information (SSI) and a shared
4 file which is shared by the accessed server and the requesting client terminal.
5 Processor 101 and browser 102 can individually control a communication
6 device, or interface 104 to access one of the servers directly or via a proxy
7 server in the network. Each server 120 includes an interface 121 connected to
8 the network 110, a processor 122 and a memory device 123 for storing the
9 server specific information of its own server and a shared file. A differential
10 data management unit 124 is connected to the processor 122 for producing
11 differential data. Processor 122 receives HTTP protocol data packet from the
12 network 110 via the interface 121 and takes an appropriate action depending
13 on the type of each packet.

14 The server specific information stored in the memory device 123
15 consists of a server address (such as www. server), a server port number
16 (such as 8080), and a variety of server-dependent information which are
17 shared between the server and the client such as download URLs
18 (http://www. server: 8080, for example) and a transfer status flag indicating
19 whether the shared data file is transferred to another server.

20 The operation of the client-server system of this invention proceeds
21 according to the flowchart of Fig. 2.

22 When the user opens a Web page and enters an URL. (step 200), the
23 browser 102 sends an HTTP request packet A1 (see Fig. 1) to the network 110.
24 If the request specifies the Web server 120-1, the packet may be routed
25 through the proxy server 111 to the server 210-1 (step 201). The URL of the

TOE50 60802860

NE-1050

- 6 -

1 request packet A1 is http://www. server/datafile. dat, for example. In
2 response, the accessed server sends its server specific information (SSI) to the
3 requesting browser. Browser 102 then checks to see if the server specific
4 information is received from the cache of the proxy server (step 202). If the
5 server specific information of the server 120-1 is previously fetched by the
6 browser 102 or by any other browser in the network and is stored in the cache
7 (step 202), the browser 102 receives the SSI from the proxy server (step 203).
8 If no information is stored in the cache, the browser 102 receives the server
9 specific information from the server 120-1 (step 204). When the browser
10 receives the SSI, it saves it in the memory device 103 (step 205) to be
11 processed later, or proceeds to step 206 to invoke the client's processor 101 in
12 order to hand over the received SSI.

13 The server accessed by the browser may receive server specific
14 information from another (second) server which belongs to a group of servers
15 when the SSI is registered in the second server (step 215). In this case, the
16 server accessed by the browser sends the received SSI to the browser, which
17 receives it at step 204.

18 If the client's processor 101 is invoked by the browser (step 207), the
19 processor determines whether SSI is received from the browser. If SSI is
20 received, the processor 101 saves it in the memory device 103 (step 209) and
21 proceeds to step 211. If no SSI is received from the browser, the processor
22 proceeds from step 208 to step 210, recognizing that the browser has executed
23 step 205, and reads the stored SSI from the memory device 103 and proceeds
24 to step 211.

25 At step 211, the processor 101 recognizes that a Web server specified

FOIE50-60802860

NE-1050

- 7 -

1 by the SSI is the server that holds the data file to be shared and sends an
2 HTTP request packet A2 to that server. The URL of the packet A2 contains a
3 path "http://www. server/RealFile_NoCache_12ABCDEF/datafile. dat", for
4 example. Shared data file is obtained from the server by ignoring the path
5 "RealFile". Further, "NoCache" is uniquely changed for every access to the
6 server so that no hit occurs in the cache.

7 If the data file to be shared is resident in the browser-accessed server
8 120-1, the request packet A2 is routed to the server 120-1. If the shared data
9 file has moved from the server 120-1 to the server 120-2, the SSI sent in
10 response to the previous request packet A1 contains the address of the new
11 server 120-2 and the second request packet A2 is routed to this server as
12 shown in Fig. 1.

13 In response to a request packet A2, the processor 122 of the accessed
14 server 120-2 recognizes that it is targeted and reads the shared data file (SF)
15 from its memory device 123 and sends it to the network as shown in Fig. 1.
16 The requesting client's processor 101 receives the transmitted file SF and
17 saves it in the memory device 103 (step 212).

18 If the shared data file is maintained in the browser-accessed server
19 120-1, the processor 101 sends a request packet A3 to server 120-1 (step 213)
20 for requesting differential data that represents the difference between the
21 data file stored in the server and the data file stored in the client terminal.
22 The URL of the packet A3 is http://server address/differential data
23 exchange character string/shared data file. Differential data manager 124 of
24 the accessed server responds to this request and transmits differential data to
25 the client terminal. The requesting processor downloads the transmitted data

TOTAL 60802860

NE-1050

- 8 -

1 into the memory device 103 to update the stored data file (step 214). If the
2 shared data file has been moved to the server 120-2 as described above, the
3 processor 101 sends a request packet A3 to the server 120-2 to request the
4 differential data (DD) as indicated in Fig. 1.

5 It is seen from the foregoing that the location of a shared file can be
6 changed from one server to another by simply updating the server specific
7 information by changing its server address from the old to the new one.
8 Since the server specific information is sent whenever a server is accessed
9 from a client terminal, the client terminal knows the new location of the
10 shared file and obtains its URL to access the new server on a real-time basis.
11 In addition, there is no need to alter the contents of data file when the data is
12 moved from one server to another.

13 Further, if the server specific information is saved in the memory
14 device 103 which the browser has received from the server it has accessed
15 (step 205), the saved information can be used by the processor at a later time
16 in a subsequent access to the network in so far as the server has not changed
17 the information content, since the process of acquisition of server specific
18 information by the browser and the process of acquisition of a shared data
19 file by the processor are clearly separated from each other.

T01E50: 608D/860